SIAM J. DISCRETE MATH.
Vol. 12, No. 2, pp. 276–287

INDEPENDENT SETS IN ASTEROIDAL TRIPLE-FREE GRAPHS*

HAJO BROERSMA†, TON KLOKS‡, DIETER KRATSCH§, AND HAIKO MÜLLER§

Abstract. An asteroidal triple (AT) is a set of three vertices such that there is a path between any pair of them avoiding the closed neighborhood of the third. A graph is called AT-free if it does not have an AT. We show that there is an $O(n^4)$ time algorithm to compute the maximum weight of an independent set for AT-free graphs. Furthermore, we obtain $O(n^4)$ time algorithms to solve the INDEPENDENT DOMINATING SET and the INDEPENDENT PERFECT DOMINATING SET problems on AT-free graphs. We also show how to adapt these algorithms such that they solve the corresponding problem for graphs with bounded asteroidal number in polynomial time. Finally, we observe that the problems CLIQUE and PARTITION INTO CLIQUES remain NP-complete when restricted to AT-free graphs.

Key words. graph algorithms, AT-free graphs, independent set, independent dominating set

AMS subject classifications. 68R10, 05C85

PII. S0895480197326346

1. Introduction.

Asteroidal triples (ATs) were introduced in 1962 to characterize interval graphs as those chordal graphs that do not contain an AT [20]. Graphs not containing an AT are called AT-free graphs. They form a large class of graphs containing interval, permutation, trapezoid, and cocomparability graphs. Since 1989, AT-free graphs have been studied extensively by Corneil, Olariu, and Stewart. They have published a collection of papers presenting many structural and algorithmic properties of AT-free graphs (see, e.g., [6, 7]). Further results on AT-free graphs were obtained in [2, 18, 23].

Until now, knowledge of the algorithmic complexity of NP-complete graph problems when restricted to AT-free graphs was relatively small compared to that of other graph classes. The problems TREEWIDTH, PATHWIDTH, and MINIMUM FILL-IN remain NP-complete on AT-free graphs [1, 28]. On the other hand, various domination-type problems like CONNECTED DOMINATING SET [2, 7], CARDINALITY STEINER TREE [2], DOMINATING SET [19], and TOTAL DOMINATING SET [19] can be solved by polynomial time algorithms for AT-free graphs. However, there is a collection of classical NP-complete graph problems for which the algorithmic complexity when restricted to AT-free graphs was not known. Prominent representatives are INDEPENDENT SET, CLIQUE, GRAPH k-COLORABILITY, PARTITION INTO CLIQUES, HAMILTONIAN CIRCUIT, and HAMILTONIAN PATH.

A crucial reason for the lack of progress in designing efficient algorithms for NP-complete problems on AT-free graphs seemed to be that none of the typical representations, which are useful for the design of efficient algorithms on special graph classes, are known for AT-free graphs. Contrary to well-known graph classes such as chordal, permutation, and circular-arc graphs, no geometric representation of AT-free

*Received by the editors August 20, 1997; accepted for publication (in revised form) August 11, 1998; published electronically April 29, 1999.
http://www.siam.org/journals/sidma/12-2/32634.html

†University of Twente, Faculty of Applied Mathematics, P.O. Box 217, 7500 AE Enschede, the Netherlands (H.J.Broersma@math.utwente.nl).

‡Department of Applied Mathematics and DIMATIA, Charles University, Malostranské nám. 25, 11800 Praha 1, Czech Republic (ton@kam.ms.mff.cuni.cz).

§Friedrich-Schiller-Universität Jena, Fakultät für Mathematik und Informatik, 07740 Jena, Germany (kratsch@minet.uni-jena.de, hm@minet.uni-jena.de).

graphs is known. Furthermore no representation of AT-free graphs by an elimination scheme of vertices or edges, small separators, or a small number of certain separators is known. Fortunately, it turns out that the design of all our algorithms is supported by a simple structural property of AT-free graphs that can be obtained from the definition of AT-free graphs rather easily.

Our approach in this paper is similar to the one used to design algorithms for problems such as TREEWIDTH [14, 17], MINIMUM FILL-IN [17], and VERTEX RANKING [18] on AT-free graphs. However, these algorithms have polynomial running time only under the additional constraint that the number of minimal separators is bounded by a polynomial in the number of vertices of the graph. (Notice that all three problems are NP-complete on AT-free graphs.) Technically, for the three different independent set problems in this paper, we are able to replace the set of all minimal separators, used in [14, 17, 18] (which might be "too large" in size) by the "small" set of all closed neighborhoods of the vertices of the graph.

Finding out the algorithmic complexity of INDEPENDENT SET on AT-free graphs is a challenging task. Besides the fact that INDEPENDENT SET is a classical and well-studied NP-complete problem, the problem is also interesting because, contrary to well-known subclasses of AT-free graphs such as cocomparability graphs, not all AT-free graphs are perfect. Thus the polynomial time algorithm for perfect graphs of Grötschel, Lovász, and Schrijver [11] that solves the INDEPENDENT SET problem does not apply to AT-free graphs.

We present the first polynomial time algorithm solving the NP-complete problem INDEPENDENT SET, when restricted to AT-free graphs. More precisely, our main result is an $O(n^4)$ algorithm to compute the maximum weight of an independent set in an AT-free graph. Furthermore, we present $O(n^4)$ time algorithms to solve the problem INDEPENDENT DOMINATING SET and INDEPENDENT PERFECT DOMINATING SET (also called EFFICIENT DOMINATING SET). We also observe that the problems CLIQUE and PARTITION INTO CLIQUES remain NP-complete when restricted to AT-free graphs.

A natural generalization of ATs are the so-called asteroidal sets. Structural results for asteroidal sets and algorithms for graphs with bounded asteroidal number were obtained in [15, 21, 25, 27]. Computing the asteroidal number (i.e., the maximum cardinality of an asteroidal set) turns out to be NP-complete in general, but solvable in polynomial time for many graph classes [16]. Furthermore, the results for problems such as TREEWIDTH and MINIMUM FILL-IN on AT-free graphs can be generalized to graphs with bounded asteroidal number [15].

We show how to adapt our algorithms to obtain polynomial time algorithms for graphs with bounded asteroidal number solving the problems INDEPENDENT SET, INDEPENDENT DOMINATING SET, and INDEPENDENT PERFECT DOMINATING SET.

2. Preliminaries. We denote the number of vertices of a graph $G = (V, E)$ by n and the number of edges by m.

Recall that an independent set in a graph G is a set of pairwise nonadjacent vertices. The independence number of a graph G denoted by $\alpha(G)$ is the maximum cardinality of an independent set in G.

For a graph $G = (V, E)$ and $W \subseteq V$, $G[W]$ denotes the subgraph of G induced by the vertices of W; we write $\alpha(W)$ for $\alpha(G[W])$. For convenience, for a vertex x of G we write $G - x$ instead of $G[V \setminus \{x\}]$. Analogously, for a subset $X \subseteq V$, we write $G - X$ instead of $G[V \setminus X]$. We consider components of a graph as maximal connected subgraphs as well as vertex subsets. For a vertex x of $G = (V, E)$, $N(x) = \{y \in V : \{x, y\} \in E\}$ is the neighborhood of x and $N[x] = N(x) \cup \{x\}$ is the closed neighborhood of x. For $W \subseteq V$, $N[W] = \bigcup_{x \in W} N[x]$.

A set $S \subseteq V$ is a separator of the graph $G = (V, E)$ if $G - S$ is disconnected.

DEFINITION 2.1. *Let $G = (V, E)$ be a graph. A set $\Omega \subseteq V$ is an* asteroidal set *if for every $x \in \Omega$ the set $\Omega \setminus \{x\}$ is contained in one component of $G - N[x]$. An asteroidal set with three vertices is called an* AT.

Notice that every asteroidal set is an independent set.

Remark 2.1. A triple $\{x, y, z\}$ of vertices of G is an AT if and only if for every two of these vertices there is a path between them avoiding the closed neighborhood of the third.

DEFINITION 2.2. *A graph $G = (V, E)$ is called* AT-free *if G has no AT.*

It is well known that the INDEPENDENT SET problem, "Given a graph G and a positive integer k, decide whether $\alpha(G) \geq k$," is NP-complete [9]. The problem remains NP-complete, even when restricted to triangle-free, 3-connected, cubic planar graphs [26]. Moreover, the independence number is hard to approximate within a factor of $n^{1-\epsilon}$ for any constant $\epsilon > 0$ [12]. Despite this discouraging recent result on the complexity of approximation, the independence number can be computed in polynomial time on many special classes of graphs (see [13]). For example, the best known algorithm to solve the problem on cocomparability graphs has running time $O(n + m)$ (see [24]).

The main result of this paper is an $O(n^4)$ algorithm to compute the maximum weight of an independent set in an AT-free graph with real vertex weights. The structural properties enabling the design of our algorithms are given in the next three sections. For convenience, we deal with the cardinality case of our problems first, and point out how to extend the method to graphs with real vertex weights in section 9.

3. Intervals. Let $G = (V, E)$ be an AT-free graph, and let x and y be two distinct nonadjacent vertices of G. Throughout the paper we use $C^x(y)$ to denote the component of $G - N[x]$ containing y, and $r(x)$ to denote the number of components of $G - N[x]$.

DEFINITION 3.1. *A vertex $z \in V \setminus \{x, y\}$ is* between x and y *if x and z are in one component of $G - N[y]$ and y and z are in one component of $G - N[x]$.*

Equivalently, z is between x and y in G if there is an x, z-path avoiding $N[y]$ and there is an y, z-path avoiding $N[x]$.

DEFINITION 3.2. *The* interval $I = I(x, y)$ *of G is the set of all vertices of G that are between x and y.*

Thus $I(x, y) = C^x(y) \cap C^y(x)$.

4. Splitting intervals. Let $G = (V, E)$ be an AT-free graph; let $I = I(x, y)$ be a nonempty interval of G; and let $s \in I$. Let $I_1 = I(x, s)$ and $I_2 = I(s, y)$.

LEMMA 4.1. *x and y are in different components of $G - N[s]$.*

Proof. Assume x and y would be in the same component of $G - N[s]$. Then there is an x, y-path avoiding $N[s]$. However, $s \in I$ implies that there is an s, y-path avoiding $N[x]$ and an s, x-path avoiding $N[y]$. Thus $\{s, x, y\}$ is an AT of G, which is a contradiction. ∎

COROLLARY 4.2. *$I_1 \cap I_2 = \emptyset$.*

Proof. Assume $z \in I_1 \cap I_2$. Then $z \in I_1$ implies that there is a component C^s of $G - N[s]$ containing both x and z. Furthermore, $z \in I_2$ implies also that $y \in C^s$, contradicting Lemma 4.1. ∎

LEMMA 4.3. *$I_1 \subseteq I$ and $I_2 \subseteq I$.*

Proof. Let $z \in I_1$. Clearly $s \in I$ implies $s \in C^x(y)$. Thus $z \in I_1$ implies $z \in C^x(y)$. Clearly $z \in C^s(x)$ since $z \in I_1$. By Lemma 4.1, $C^s(x)$ is contained in a component of $G - N[y]$, and obviously this component contains x. This proves $z \in I$. Consequently $I_1 \subseteq I$.

$I_2 \subseteq I$ can be shown analogously. ◻

THEOREM 4.4. *There exist components $C_1^s, C_2^s, \ldots, C_t^s$ of $G - N[s]$ such that*

$$I \setminus N[s] = I_1 \cup I_2 \cup \bigcup_{i=1}^{t} C_i^s.$$

Proof. By Lemma 4.3, we have $I_1 \subseteq I \setminus N[s]$ and $I_2 \subseteq I \setminus N[s]$. By Lemma 4.1, x and y belong to different components $C^s(x)$ and $C^s(y)$ of $G - N[s]$. Let $z \in I \setminus N[s]$.

Assume $z \in C^s(x)$. There is a z, y-path avoiding $N[x]$. This path must contain a vertex of $N[s]$, showing the existence of a z, s-path avoiding $N[x]$. Hence $z \in I_1$.

Similarly $z \in C^s(y)$ implies $z \in I_2$.

Assume $z \notin C^s(x)$ and $z \notin C^s(y)$. Since $z \notin N[s]$, z belongs to the component $C^s(z)$ of $G - N[s]$. For any vertex $p \in C^s(z)$, there is a p, z-path avoiding $N[x]$, since $C^s(z) \neq C^s(x)$. Since $z \in I$, there is a z, y-path avoiding $N[x]$. Hence there is also a p, y-path avoiding $N[x]$. This shows $C^s(z) \subseteq I \setminus N[s]$. ◻

COROLLARY 4.5. *Every component of $G[I \setminus (N[s] \cup I_1 \cup I_2)]$ is a component of $G - N[s]$.*

5. Splitting components. Let $G = (V, E)$ be an AT-free graph. Let C^x be a component of $G - N[x]$ and let y be a vertex of C^x. Thus $C^x = C^x(y)$. We study the components of the graph $C^x - N[y]$.

THEOREM 5.1. *Let D be a component of the graph $C^x - N[y]$. Then $N[D] \cap (N[x] \setminus N[y]) = \emptyset$ if and only if D is a component of $G - N[y]$.*

Proof. Let D be a component of $C^x - N[y]$ with $N[D] \cap (N[x] \setminus N[y]) = \emptyset$. Since no vertex of D has a neighbor in $N[x] \setminus N[y]$, D is a component of $G - N[y]$.

Now let $D \subseteq C^x$ be a component of $G - N[y]$. Then $N[D] \cap N[x] \subseteq N[y]$. ◻

COROLLARY 5.2. *Let B be a component of the graph $C^x - N[y]$. Then $N[B] \cap (N[x] \setminus N[y]) \neq \emptyset$ if and only if $B \subseteq C^y(x)$.*

THEOREM 5.3. *Let $B_1, \ldots, B_\ell$ denote the components of the graph $C^x - N[y]$ that are contained in $C^y(x)$. Then $I(x, y) = \bigcup_{i=1}^{\ell} B_i$.*

Proof. Let $I = I(x, y)$. First we show that $B_i \subseteq I$ for every $i \in \{1, \ldots, \ell\}$. Let $z \in B_i$. There is an x, z-path avoiding $N[y]$, since some vertex in B_i has a neighbor in $N[x] \setminus N[y]$. Clearly, there is also a z, y-path avoiding $N[x]$, since z and y are both in C^x. This shows that $z \in I$. Consequently $\bigcup_{i=1}^{\ell} B_i \subseteq I$.

Suppose $z \in I \setminus \bigcup_{i=1}^{\ell} B_i$. Since $z \notin \bigcup_{i=1}^{\ell} B_i$, the component D of $C^x - N[y]$ that contains z does not contain a vertex with a neighbor in $N[x] \setminus N[y]$. Thus $z \notin C^y(x)$, implying $z \notin I$, a contradiction. ◻

6. Computing the independence number. In this section we describe our algorithm to compute the independence number of an AT-free graph. The algorithm we propose uses dynamic programming on intervals and components. All intervals and all components are sorted according to a nondecreasing number of vertices. Following this order, the algorithm determines the independence number of each component and of each interval using the formulas given in Lemmas 6.1, 6.2, and 6.3.

We start with an obvious lemma.

LEMMA 6.1. *Let $G = (V, E)$ be any graph. Then*

$$\alpha(G) = 1 + \max_{x \in V} \left(\sum_{i=1}^{r(x)} \alpha(C_i^x) \right),$$

where $C_1^x, C_2^x, \ldots, C_{r(x)}^x$ are the components of $G - N[x]$.

Applying Lemma 6.1 to the decomposition given by Theorems 5.1 and 5.3, we obtain the following lemma.

LEMMA 6.2. *Let $G = (V, E)$ be an AT-free graph. Let $x \in V$ and let C^x be a component of $G - N[x]$. Then*

$$\alpha(C^x) = 1 + \max_{y \in C^x} \left(\alpha(I(x, y)) + \sum_i \alpha(D_i^y) \right),$$

where the D_i^y's are the components of $G - N[y]$ contained in C^x.

Applying Lemma 6.1 to the decomposition given by Theorem 4.4, we obtain the following lemma.

LEMMA 6.3. *Let $G = (V, E)$ be an AT-free graph. Let $I = I(x, y)$ be an interval of G. If $I = \emptyset$ then $\alpha(I) = 0$. Otherwise*

$$\alpha(I) = 1 + \max_{s \in I} \left(\alpha(I(x, s)) + \alpha(I(s, y)) + \sum_i \alpha(C_i^s) \right),$$

where the C_i^s's are the components of $G - N[s]$ contained in $I(x, y)$.

Remark 6.1. Notice that the components D_i^y and C_i^s as well as the intervals $I(x, s)$ and $I(s, y)$ on the right-hand sides of the formulas in Lemmas 6.2 and 6.3 are proper subsets of C^x and I, respectively. Hence $\alpha(C^x)$ (respectively, $\alpha(I)$) can be computed by table look-up to components and intervals with a smaller number of vertices.

Consequently we obtain the following algorithm to compute the independence number $\alpha(G)$ for a given AT-free graph $G = (V, E)$, which is based on dynamic programming.

Step 1. For every $x \in V$ compute all components $C_1^x, C_2^x, \ldots, C_{r(x)}^x$ of $G - N[x]$.

Step 2. For every pair of nonadjacent vertices x and y compute the interval $I(x, y)$.

Step 3. Sort all the components and intervals according to nondecreasing number of vertices.

Step 4. Compute $\alpha(C)$ and $\alpha(I)$ for each component C and each interval I in the order of Step 3.

Step 5. Compute $\alpha(G)$.

THEOREM 6.4. *There is an $O(n^4)$ time algorithm to compute the independence number of a given AT-free graph.*

Proof. The correctness of the algorithm follows from the formulas of Lemmas 6.1, 6.2, and 6.3 as well as the order of the dynamic programming.

We show how to obtain the stated time complexity. Clearly, Step 1 can be implemented such that it takes $O(n(n+m))$ time using a linear time algorithm to compute the components of the graph $G - N[x]$ for each vertex x of G. For each component of $G - N[x]$, a sorted linked list of all its vertices and its number of vertices is stored. For all nonadjacent vertices x and y there is a pointer $P(x, y)$ to the list of $C^x(y)$. Thus in Step 2, an interval $I(x, y)$ can be computed using the fact that

$I(x,y) = C^x(y) \cap C^y(x)$. Hence a sorted vertex list of $I(x,y)$ can be computed in time $O(n)$ for each interval. Consequently the overall time bound for Step 2 is $O(n^3)$. There are at most n^2 components and at most n^2 intervals and each has at most n vertices. Thus, using the linear time sorting algorithm bucket sort, Step 3 can be done in time $O(n^2)$.

The bottleneck for the time complexity of our algorithm is Step 4. First consider a component C^x of $G-N[x]$ and a vertex $y \in C^x$. We need to compute the components of $G-N[y]$ that are contained in C^x. Each component D of $G-N[y]$ except $C^y(x)$ is contained in C^x if and only if $D \cap C^x \neq \emptyset$. Thus the components D of $G-N[y]$ with $D \subseteq C^x$ are exactly those components of $G-N[y]$ addressed by $P(y,z)$ for some $z \in C^x$. Thus all such components can be found in time $O(|C^x|)$ for fixed vertices x and $y \in C^x$. Hence the computation of $\alpha(C)$ for all components C takes time $\sum_{\{x,y\}\notin E} O(|C^x(y)|) = O(n^3)$.

Now consider an interval $I = I(x,y)$, and a vertex $s \in I$. We need to add up the independence numbers of the components C_i^s of $G-N[s]$ that are contained in I. The components of $G-N[y]$ that are contained in I are exactly those components addressed by $P(y,z)$ for some $z \in I$, except $C^s(x)$ and $C^s(y)$. Thus all such components can be found in time $O(|I(x,y)|)$ for a fixed interval $I(x,y)$ and $s \in I(x,y)$. Hence the computation of $\alpha(I)$ for all intervals I takes time $\sum_{\{x,y\}\notin E} \sum_{s\in I(x,y)} O(|I(x,y)|) = O(n^4)$.

Clearly Step 5 can be done in $O(n^2)$ time. Thus the running time of our algorithm is $O(n^4)$. $\square$

7. Independent domination. The approach used to design the presented polynomial time algorithm to compute the independence number for AT-free graphs can also be used to obtain a polynomial time algorithm solving the INDEPENDENT DOMINATING SET problem on AT-free graphs. The best known algorithm to solve the weighted version of the problem on cocomparability graphs has running time $O(n^{2.376})$ [4].

DEFINITION 7.1. *Let $G=(V,E)$ be a graph. Then $S \subseteq V$ is a* dominating set *of G if every vertex of $V \setminus S$ has a neighbor in S. A dominating set $S \subseteq V$ is an* independent dominating set *of G if S is an independent set.*

We denote by $\gamma_{\mathrm{i}}(G)$ the minimum cardinality of an independent dominating set of the graph G. Given an AT-free graph G, our next algorithm computes $\gamma_{\mathrm{i}}(G)$. It works very similarly to the algorithm of the previous section.

We present only the formulas used in Steps 4 and 5 of the algorithm (which are similar to those in Lemmas 6.1, 6.2, and 6.3).

LEMMA 7.2. *Let $G=(V,E)$ be a graph. Then*

$$\gamma_{\mathrm{i}}(G) = 1 + \min_{x\in V}\left(\sum_{j=1}^{r(x)} \gamma_{\mathrm{i}}(C_j^x)\right),$$

where $C_1^x, C_2^x, \ldots, C_{r(x)}^x$ are the components of $G-N[x]$.

LEMMA 7.3. *Let $G=(V,E)$ be an AT-free graph. Let $x \in V$ and let C^x be a component of $G-N[x]$. Then*

$$\gamma_{\mathrm{i}}(C^x) = 1 + \min_{y\in C^x}\left(\gamma_{\mathrm{i}}(I(x,y)) + \sum_j \gamma_{\mathrm{i}}(D_j^y)\right),$$

where the D^y_j's are the components of $G - N[y]$ contained in C^x.

LEMMA 7.4. *Let $G = (V, E)$ be an AT-free graph. Let $I = I(x, y)$ be an interval. If $I = \emptyset$ then $\gamma_{\mathrm{i}}(I) = 0$. Otherwise*

$$\gamma_{\mathrm{i}}(I) = 1 + \min_{s \in I} \left(\gamma_{\mathrm{i}}(I(x, s)) + \gamma_{\mathrm{i}}(I(s, y)) + \sum_j \gamma_{\mathrm{i}}(C^s_j) \right),$$

where the C^s_j's are the components of $G - N[s]$ contained in $I(x, y)$.

The design and analysis of the algorithm is done similarly to the one in the previous section. This gives the following theorem.

THEOREM 7.5. *There exists an $O(n^4)$ time algorithm to compute the independence domination number γ_{i} of a given AT-free graph.*

8. Independent perfect domination. The INDEPENDENT PERFECT DOMINATING SET problem is a variant of the INDEPENDENT DOMINATING SET problem. The best known algorithm to solve the weighted version of the problem on cocomparability graphs has running time $O(n^2)$ [5].

DEFINITION 8.1. *A* perfect dominating set *of a graph $G = (V, E)$ is a set $S \subseteq V$ such that every vertex of $V \setminus S$ is adjacent to exactly one vertex in S. A perfect dominating set S is an* independent perfect dominating set *of G if S is an independent set. (An independent perfect dominating set is also called an* efficient dominating set.*)*

We denote the minimum cardinality of an independent perfect dominating set in G by $\gamma_{\mathrm{ip}}(G)$. If G does not have an independent perfect dominating set, we define $\gamma_{\mathrm{ip}}(G) = \infty$.

There is a close relationship between the problems INDEPENDENT PERFECT DOMINATING SET and INDEPENDENT DOMINATING SET which can often be exploited to transform an algorithm solving the INDEPENDENT DOMINATING SET problem into an algorithm solving the INDEPENDENT PERFECT DOMINATING SET problem. We demonstrate this for our algorithm of the previous section.

We present the formulas for an $O(n^4)$ algorithm to compute $\gamma_{\mathrm{ip}}(G)$ for a given AT-free graph G. Let x be a vertex of G and let C^x be a component of $G - N[x]$. Let $\Delta(x, C^x) = \{z \in C^x \mid d_G(z, x) > 2\}$. We denote by $\gamma_{\mathrm{ip}}(x, C^x)$ the minimum cardinality of an independent perfect dominating set S of C^x with $S \subseteq \Delta(x, C^x)$.

LEMMA 8.2. *Let $G = (V, E)$ be a graph. Then*

$$\gamma_{\mathrm{ip}}(G) = 1 + \min_{x \in V} \left(\sum_{i=1}^{r(x)} \gamma_{\mathrm{ip}}(x, C^x_i) \right),$$

where $C^x_1, C^x_2, \ldots, C^x_{r(x)}$ are the components of $G - N[x]$.

Let $I = I(x, y)$ be an interval of G. Let $\Delta(x, y, I) = \{z \in I \mid d_G(z, x) > 2 \ \wedge \ d_G(z, y) > 2\}$. We denote by $\gamma_{\mathrm{ip}}(x, y, I(x, y))$ the minimum cardinality of an independent perfect dominating set S of $G[I(x, y)]$ with $S \subseteq \Delta(x, y, I)$.

LEMMA 8.3. *Let $G = (V, E)$ be an AT-free graph. Let $x \in V$ and let C^x be a component of $G - N[x]$. If $\Delta(x, C^x) = \emptyset$ then $\gamma_{\mathrm{ip}}(x, C^x) = \infty$. If $\Delta(x, C^x) \neq \emptyset$ then*

$$\gamma_{\mathrm{ip}}(x, C^x) = 1 + \min_{y \in \Delta(x, C^x)} \left(\gamma_{\mathrm{ip}}(x, y, I(x, y)) + \sum_j \gamma_{\mathrm{ip}}(y, D^y_j) \right),$$

where the D_j^y's are the components of $G - N[y]$ contained in C^x.

LEMMA 8.4. *Let $G = (V, E)$ be an AT-free graph. Let $I(x, y)$ be an interval of G. For $s \in I$, let $I_1 = I(s, x)$ and $I_2 = I(s, y)$. If $\Delta(x, y, I) = \emptyset$ and $|I| = 0$ then $\gamma_{\mathrm{ip}}(x, y, I) = 0$. If $\Delta(x, y, I) = \emptyset$ and $|I| > 0$ then $\gamma_{\mathrm{ip}}(x, y, I) = \infty$. If $\Delta(x, y, I) \neq \emptyset$ then*

$$\gamma_{\mathrm{ip}}(x, y, I) = 1 + \min_{s \in \Delta(x,y,I)} \left(\gamma_{\mathrm{ip}}(x, s, I_1) + \gamma_{\mathrm{ip}}(s, y, I_2) + \sum_j \gamma_{\mathrm{ip}}(s, C_j^s) \right),$$

where the C_j^s's are the components of $G - N[s]$ contained in $I(x, y)$.

Our algorithm first computes the distance matrix of the given graph and then applies the approach of the previous two sections.

THEOREM 8.5. *There exists an $O(n^4)$ time algorithm to compute the independent perfect domination number γ_{ip} of a given AT-free graph.*

9. Weights on the vertices. In this section we consider AT-free graphs with real weights. Since we assume a unit-cost RAM as computational model, weights can be compared and added in constant time.

DEFINITION 9.1. *A* weighted *graph is a pair (G, w), where $G = (V, E)$ is a graph and every vertex x of G is assigned a real weight $w(x)$. Let $S \subseteq V$. Then $w(S) = \sum_{x \in S} w(x)$ is the weight of S.*

For a weighted graph (G, w) the maximum weight of an independent set of G is denoted by $\alpha^w(G)$, and the minimum weight of an independent dominating set of G is denoted by $\gamma_{\mathrm{i}}^w(G)$. Clearly, $\alpha^w(G) = \alpha^w(G[\{x \in V\colon\ w(x) > 0\}])$.

First we prove a version of Lemma 6.1 extended to weighted graphs.

LEMMA 9.2. *Let (G, w) be a weighted graph, $G = (V, E)$. Then*

$$\alpha^w(G) = \max_{x \in V, w(x) > 0} \left(w(x) + \sum_{i=1}^{r(x)} \alpha^w(C_i^x) \right),$$

where $C_1^x, C_2^x, \ldots, C_{r(x)}^x$ are the components of $G - N[x]$.

Proof. If $w(x) \leq 0$ for all $x \in V$ then $\alpha^w(G) = 0$, since the empty set is independent. Otherwise G has a nonempty independent set S of maximum weight containing vertices of positive weight only. For such a set we have $x \in S$ if and only if $w(S) = w(x) + \sum_{i=1}^{r(x)} \alpha^w(C_i^x)$, where $C_1^x, C_2^x, \ldots, C_{r(x)}^x$ are the components of $G - N[x]$. ◻

The two remaining lemmas of section 6 generalize to weighted AT-free graphs in a similar way. We obtain the formulas

$$\alpha^w(C^x) = \max_{y \in C^x, w(y) > 0} \left(w(y) + \alpha^w(I(x, y)) + \sum_i \alpha^w(D_i^y) \right),$$

$$\alpha^w(I) = \max_{s \in I, w(s) > 0} \left(w(s) + \alpha^w(I(x, s)) + \alpha^w(I(s, y)) + \sum_i \alpha^w(C_i^s) \right),$$

analogously to the formulas in Lemmas 6.2 and 6.3, respectively. Therefore, the algorithm given in section 6 applied to a weighted AT-free graph computes the maximum weight of an independent set and runs in time $O(n^4)$.

For the problem INDEPENDENT DOMINATING SET on weighted AT-free graphs, we obtain the formulas

$$\gamma_{\mathrm{i}}^{w}(G) = \min_{x \in V} \left(w(x) + \sum_{j=1}^{r(x)} \gamma_{\mathrm{i}}^{w}(C_j^x) \right),$$

$$\gamma_{\mathrm{i}}^{w}(C^x) = \min_{y \in C^x} \left(w(y) + \gamma_{\mathrm{i}}^{w}(I(x,y)) + \sum_{j} \gamma_{\mathrm{i}}^{w}(D_j^y) \right),$$

$$\gamma_{\mathrm{i}}^{w}(I) = \min_{s \in I} \left(w(s) + \gamma_{\mathrm{i}}^{w}(I(x,s)) + \gamma_{\mathrm{i}}^{w}(I(s,y)) + \sum_{j} \gamma_{\mathrm{i}}^{w}(C_j^s) \right),$$

analogously to the formulas in Lemmas 7.2, 7.3, and 7.4, respectively. Consequently, there exists an algorithm computing $\gamma_{\mathrm{i}}^{w}(G)$ for a weighted AT-free graph G in time $O(n^4)$.

10. Bounded asteroidal number. In this section we show that the independence number of graphs with bounded asteroidal number can be computed in polynomial time.

DEFINITION 10.1. *The* asteroidal number *of a graph G is the maximum cardinality of an asteroidal set in G.*

Hence a graph is AT-free if and only if its asteroidal number is at most 2. Furthermore, the asteroidal number of a graph G is bounded by $\alpha(G)$, since every asteroidal set is an independent set. Computing the asteroidal number of a graph is NP-complete in general, but solvable in polynomial time for many graph classes [16].

DEFINITION 10.2. *Let Ω be an asteroidal set of G. The* lump $L(\Omega)$ *is the set of vertices v such that for all $x \in \Omega$ there is a component of $G - N[x]$ containing v and $\Omega \setminus \{x\}$.*

Let $\Omega = \{x_1, \ldots, x_\kappa\}$ be an asteroidal set of cardinality $\kappa \geq 2$ and consider the lump $L = L(\Omega)$.

Let s be an arbitrary vertex in L. Now we show how $N[s]$ splits the lump analogously to Theorem 4.4.

Consider the components of $G - N[s]$. These components partition Ω into sets $\Omega_1, \ldots, \Omega_\tau$, where each Ω_i is a maximal subset of Ω contained in a component of $G - N[s]$.

LEMMA 10.3. *For each $i = 1, \ldots, \tau$, the set $\Omega_i^* = \Omega_i \cup \{s\}$ is an asteroidal set in G.*

Proof. Consider $x \in \Omega_i$. Then, by definition, $\Omega \setminus \{x\}$ and s are contained in one component of $G - N[x]$. Hence, $\Omega_i^* \setminus \{x\}$ is contained in one component of $G - N[x]$. This proves the claim. ∎

LEMMA 10.4. *Let $z \in L$ be in some component C^* of $G - N[s]$ that contains no vertices of Ω. Then $C^* \subseteq L$.*

Proof. Let $p \in C^* \setminus \{z\}$. There is a p, z-path avoiding $N[x]$ for any vertex $x \in \Omega$. This proves the claim. ∎

First we consider the case where $\tau = 1$, i.e., where Ω is in one component of $G - N[s]$. Then $\Omega \cup \{s\}$ is an asteroidal set.

LEMMA 10.5. *If Ω is contained in one component C of $G - N[s]$, then $L(\Omega \cup \{s\}) = L \cap C$.*

Proof. Clearly $L(\Omega \cup \{s\}) \subseteq L \cap C$. Let $z \in L \cap C$ and consider a vertex $x \in \Omega$. Clearly, there is an x, z-path avoiding $N[s]$, since z and x are in the component C of $G - N[s]$. Hence z is in the component of $G - N[s]$ containing Ω. Consider any other vertex $y \in \Omega$. (Such a vertex exists since $|\Omega| \geq 2$.) Then there is a z, y-path avoiding $N[x]$ since $z \in L$. Furthermore, there is a y, s-path avoiding $N[x]$ since $\Omega \cup \{s\}$ is an asteroidal set. Hence z is in the component of $(\Omega \cup \{s\}) \setminus \{x\}$ of $G - N[x]$. ∎

Now we consider the case where $\tau > 1$. Let $L_i = L(\Omega_i \cup \{s\})$ for $i = 1, \ldots, \tau$. Clearly, $L_i \cap L_j = \emptyset$ for every $i \neq j$.

LEMMA 10.6. *Assume $\tau > 1$ and let C be the component of $G - N[s]$ containing Ω_i. Then $L_i = L \cap C$.*

Proof. First let $z \in L \cap C$. Then for all x and y in Ω_i there is a z, x-path avoiding $N[s]$ since $z \in C$ (showing that z and Ω_i are in one component of $G - N[s]$), and there is a z, x-path avoiding $N[y]$ since $z \in L$. For $y' \in \Omega_j$ for any $j \neq i$ there is a z, y'-path avoiding $N[x]$, since $z \in L$. Such a path contains a vertex of $N[s]$, and consequently there is a z, s-path avoiding $N[x]$. This shows that z, s and $\Omega_i \setminus \{x\}$ are in one component of $G - N[x]$ and hence $L \cap C \subseteq L_i$.

Now let $z \in L_i$. This clearly implies $z \in C$. For a vertex $y \in \Omega_j$, $j \neq i$, s and the set $\Omega \setminus \{y\}$ are in one component of $G - N[y]$ since $s \in L$. There is an s, y-path avoiding $N[y]$ since y and z belong to different components of $G - N[s]$. Consequently, z and $\Omega \setminus \{y\}$ are in one component of $G - N[y]$.

For a vertex $x \in \Omega_i$, there is a component of $G - N[x]$ containing s and $\Omega \setminus \{x\}$, since $s \in L$. Since $z \in L_i$, there is an s, z-path avoiding $N[x]$. Hence also z is in this component of $G - N[x]$ and therefore $L_i \subseteq L \cap C$. ∎

THEOREM 10.7. *There exist components $C_1, \ldots, C_t$ of $G - N[s]$ which contain no vertex of Ω such that*

$$L \setminus N[s] = \bigcup_{i=1}^{t} C_i \cup \bigcup_{j=1}^{\tau} L_j.$$

Proof. Let $C_1, \ldots, C_t$ be the components of $G - N[s]$ which contain a vertex of L but no vertex of Ω. Then by Lemma 10.4 we have $\bigcup_{i=1}^{t} C_i \subseteq L \setminus N[s]$, and by Lemmas 10.5 and 10.6 we have $\bigcup_{j=1}^{\tau} L_j \subseteq L \setminus N[s]$.

Now let $l \in L \setminus N[s]$. If l is in a component containing Ω_i, $1 \leq i \leq \tau$, then $l \in L_i$ by Lemma 10.5 or 10.6. Otherwise there is an index i, $1 \leq i \leq t$, such that $l \in C_i$. This completes the proof. ∎

Theorem 10.7 enables us to generalize Lemmas 6.3, 7.4, and 8.4 in the following way.

LEMMA 10.8. *Let $L = L(\Omega)$ be a lump of G. If $L = \emptyset$ then $\alpha(L) = \gamma_{\mathrm{i}}(L) = \gamma_{\mathrm{ip}}(\Omega, L) = 0$. Otherwise*

$$\alpha(L) = 1 + \max_{s \in L} \left(\sum_{j=1}^{t} \alpha(C_j) + \sum_{i=1}^{\tau} \alpha(L_i) \right),$$

$$\gamma_{\mathrm{i}}(L) = 1 + \min_{s \in L} \left(\sum_{j=1}^{t} \gamma_{\mathrm{i}}(C_j) + \sum_{k=1}^{\tau} \gamma_{\mathrm{i}}(L_k) \right),$$

$$\gamma_{\mathrm{ip}}(\Omega, L) = 1 + \min_{s \in \Delta(\Omega, L)} \left(\sum_{j=1}^{t} \gamma_{\mathrm{ip}}(s, C_j) + \sum_{k=1}^{\tau} \gamma_{\mathrm{i}}(\Omega_k + s, L_k) \right),$$

where $C_1, \ldots, C_t$ are the components of $G - N[s]$ which contain no vertex of Ω, $L_1, \ldots, L_\tau$ are the lumps $L(\Omega_i + s)$ as used in Lemma 10.3*, $\Delta(\Omega, L) = \{z \in L \mid d_G(z, \Omega) > 2\}$, and $\gamma_{\mathrm{ip}}(\Omega, L)$ is the minimum cardinality of an independent perfect dominating set of $G[L]$ contained in $\Delta(\Omega, L)$. Moreover, if $L \neq \emptyset$ but $\Delta(\Omega, L) = \emptyset$, then $\gamma_{\mathrm{ip}}(\Omega, L) = \infty$.*

Together with Lemmas 6.1 and 6.2, 7.2 and 7.3, and 8.2 and 8.4, the formulas of Lemma 10.8 lead to algorithms computing $\alpha(G)$, $\gamma_{\mathrm{i}}(G)$, and $\gamma_{\mathrm{ip}}(G)$ for a graph G. For any positive integer k, these algorithms can be implemented to run in time $O(n^{k+2})$ for all graphs with asteroidal number at most k. Analogously to the proof of Theorem 6.4, the time complexity is now dominated by the term $\sum_{\Omega} \sum_{s \in L(\Omega)} O(|L(\Omega)|) = O(n^{k+2})$, where the sum is taken over all asteroidal sets Ω of G and all $s \in L(\Omega)$.

As before, our algorithms for graphs with a bounded asteroidal number can be extended to the weighted cases of the problems and the corresponding algorithms run within the same time bounds.

11. Cliques. Contrary to the independent set problems considered so far, the NP-complete graph problems CLIQUE and PARTITION INTO CLIQUES, that are closely related to INDEPENDENT SET, both remain NP-complete when restricted to the class of AT-free graphs. Concerning CLIQUE, recall that Poljak has shown that INDEPENDENT SET remains NP-complete on triangle-free graphs (see [9]). Consequently CLIQUE remains NP-complete on graphs with independence number at most 2, and thus on AT-free graphs.

Similarly, it follows from a recent result due to Maffray and Preissman (showing that the problem GRAPH k-COLORABILITY remains NP-complete when restricted to triangle-free graphs [22]), that the problem PARTITION INTO CLIQUES remains NP-complete on AT-free graphs.

Therefore CLIQUE and PARTITION INTO CLIQUES are the first NP-complete graph problems known to us which are NP-complete on AT-free graphs, but solvable in polynomial time on the class of cocomparability graphs. The latter graph class is the largest well-studied subclass of AT-free graphs which is also a class of perfect graphs.

12. Conclusions. In this paper we have shown that the maximum weight of an independent set in a weighted AT-free graph can be computed in time $O(n^4)$. The same approach can be used to obtain $O(n^4)$ algorithms to solve the (weighted) INDEPENDENT DOMINATING SET problem and the INDEPENDENT PERFECT DOMINATING SET problem on AT-free graphs. We have also shown how to adapt the algorithm computing the independence number in such a way that the new algorithm computes the independence number of a graph with a bounded asteroidal number in polynomial time.

All our algorithms can be modified such that they not only compute the optimal weight of a set of certain type (e.g., the maximum weight of an independent set) but also a set realizing the optimal weight (e.g., a maximum weight independent set) within the same time bound.

From the current knowledge it would be interesting to find out the algorithmic complexity of the following well-known NP-complete graph problems when restricted to AT-free graphs: GRAPH k-COLORABILITY, HAMILTONIAN CIRCUIT, HAMILTONIAN PATH. These three problems are all known to have polynomial time algorithms for cocomparability graphs [8, 10].

REFERENCES

[1] S. Arnborg, D. G. Corneil, and A. Proskurowski, *Complexity of finding embeddings in a k-tree*, SIAM J. Algebraic Discrete Methods, 8 (1987), pp. 277–284.

[2] H. Balakrishnan, A. Rajaraman, and C. Pandu Rangan, *Connected domination and Steiner set on asteroidal triple-free graphs*, in Proceedings of WADS'93, Lecture Notes in Computer Science 709, Springer-Verlag, New York, 1993, pp. 131–141.

[3] A. Brandstädt, *Special graph classes – A survey*, Schriftenreihe des Fachbereichs Mathematik, SM-DU-199, Universität Duisburg Gesamthochschule, Duisburg, Germany, 1991.

[4] H. Breu and D. G. Kirkpatrick, *Algorithms for Domination and Steiner Tree Problems in Cocomparability Graphs*, University of British Columbia, Vancouver, manuscript 1993.

[5] M. S. Chang, *Weighted domination on cocomparability graphs*, in Proceedings of ISAAC'95, Lecture Notes in Computer Science 1004, Springer-Verlag, New York, 1996, pp. 122–131.

[6] D. G.Corneil, S. Olariu, and L. Stewart, *Asteroidal triple-free graphs*, SIAM J. Discrete Math., 10 (1997), pp. 399–430.

[7] D. G. Corneil, S. Olariu, and L. Stewart, *A linear time algorithm to compute dominating pairs in asteroidal triple-free graphs*, in Proceedings of ICALP'95, Lecture Notes in Computer Science 944, Springer-Verlag, New York, 1995, pp. 292–302.

[8] J. S. Deogun and G. Steiner, *Polynomial algorithms for Hamiltonian cycle in cocomparability graphs*, SIAM J. Comput., 23 (1994), pp. 520–552.

[9] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, 1979.

[10] M. C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York, 1980.

[11] M. Grötschel, L. Lovász, and A. Schrijver, *Polynomial algorithms for perfect graphs*, Ann. Disc. Math., 21 (1984), pp. 325–356.

[12] J. Hastad, *Clique is hard to approximate within* $n^{1-\epsilon}$, in Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science, Burlington, VT, 1996, pp. 627–636.

[13] D. S. Johnson, *The NP-completeness column: An ongoing guide*, J. Algorithms, 6 (1985), pp. 434–451.

[14] T. Kloks, *Treewidth – Computations and Approximations*, Lecture Notes in Computer Science 842, Springer-Verlag, New York, 1994.

[15] H. J. Broersma, T. Kloks, D. Kratsch, and H. Müller, *A generalization of AT-free graphs and a generic algorithm for solving triangulation problems*, in Proceedings of WG'98, Lecture Notes in Computer Science 1517, Springer-Verlag, New York, 1998, pp. 88–89.

[16] T. Kloks, D. Kratsch, and H. Müller, *Asteroidal sets in graphs*, in Proceedings of WG'97, Lecture Notes in Computer Science 1335, Springer-Verlag, New York, 1997, pp. 229–241.

[17] T. Kloks, D. Kratsch, and J. Spinrad, *On treewidth and minimum fill-in of asteroidal triple-free graphs*, Theoret. Comput. Sci., 175 (1997), pp. 309–335.

[18] T. Kloks, H. Müller, and C. K. Wong, *Vertex ranking of asteroidal triple-free graphs*, in Proceedings of ISAAC'96, Lecture Notes in Computer Science 1178, Springer-Verlag, New York, 1996, pp. 174–182.

[19] D. Kratsch, *Domination and Total Domination on Asteroidal Triple-Free Graphs*, Forschungsergebnisse Math/Inf/96/25, FSU Jena, Germany, 1996.

[20] C. G. Lekkerkerker and J. Ch. Boland, *Representation of a finite graph by a set of intervals on the real line*, Fund. Math., 51 (1962), pp. 45–64.

[21] I. J. Lin, T. A. McKee, and D. B. West, *Leafage of chordal graphs*, Discuss. Math. Graph Theory, 18 (1998), pp. 23–48.

[22] F. Maffray and M. Preissman, *On the NP-completeness of the k-colorability problem for triangle-free graphs*, Discrete Math., 162 (1996), pp. 313–317.

[23] R. H. Möhring, *Triangulating graphs without asteroidal triples*, Discrete Appl. Math., 64 (1996), pp. 281–287.

[24] R. M. McConnell and J. P. Spinrad, *Linear time transitive orientation*, in Proceedings of the Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, Philadelphia, PA, 1997, pp. 19–25.

[25] E. Prisner, *Representing triangulated graphs in stars*, Abh. Math. Sem. der Univ. Hamburg, 62 (1992), pp. 29–41.

[26] R. Uehara, *NP-Complete Problems on a 3-Connected Cubic Planar Graph and Their Applications*, Technical report TWCU-M-0004, Tokyo Woman's Christian University, 1996.

[27] J. R. Walter, *Representations of chordal graphs as subtrees of a tree*, J. Graph Theory, 2 (1978), pp. 265–267.

[28] M. Yannakakis, *Computing the minimum fill-in is NP-complete*, SIAM J. Algebraic Discrete Methods, 2 (1981), pp. 77–79.